A Stochastic Scheduler for Integrated Arrival, Departure, and Surface Operations in Los Angeles

Min Xue*
University of California at Santa Cruz, Moffett Field, CA 94035
Shannon Zelinski†
NASA Ames Research Center, Moffett Field, CA 94035

In terminal airspace, integrating arrivals, departures, and surface operations with competing resources provides the potential of improving operational efficiency by removing barriers between different operations. This work develops a centralized stochastic scheduler for operations in a terminal area including airborne and surface operations using a non-dominated sorting genetic algorithm and Monte Carlo simulations. The scheduler handles competing resources between different flows, such as runway allocations, runway crossings, merges at departure fixes, and other interaction waypoints between arrivals and departures. The scheduler takes time-varied uncertainties into account in optimization as well. The scheduler is run sequentially to identify the best robust schedule for the next planning window. Resulting schedules determine routes, speeds or delays, and runway assignments subject to separation constraints at merging/diverging waypoints in the air and at runways (including runway crossings) on the surface. The Los Angeles terminal area was used as an example in experiments with a four-hour traffic scenario. The results showed that using stochastic schedulers can reduce flight time delay (airborne and ground) anywhere from 28% to 40% statistically compared to deterministic schedulers. Sensitivity studies on various planning horizons presented that trade-offs exist between planning horizons and achievable minimum delays. A twenty-minute planning horizon was found to be a bad choice because uncertainties increased with the look-ahead time. Eight minutes was promising for planning as it achieved the lowest delay compared to others. However, the results demonstrated that any duration from two minutes to eight minutes could be a good candidate as well. The results on runway usage showed that using the stochastic scheduler, runway makespans and occupancy were usually slightly lower than applying deterministic schedulers.

I. Introduction

In the National Airspace System (NAS) terminal areas, thousands of flights have to depart, arrive, or taxi within short periods of time in the crowded airspace/surface areas every day. High-density operations impose complexity and inefficiency on terminal airspace operations, and form choke points in the system. The situation gets severe at major airports or metroplexes. Improving the operational efficiency in terminal areas is critical for an efficient air traffic system.

In the past decade, research has been conducted from different perspectives to improve the efficiency of operations in terminal areas. There have been many studies on arrival scheduling, departure scheduling, and surface scheduling.[1–7] This research focused on aircraft sequencing problems and used airborne speed or ground push-back time control. Representative tools that were developed by NASA are: the Traffic Management Advisor (TMA)[8,9] for arrival flight scheduling by adjusting speeds; and the Spot and Runway Departure Advisor (SARDA)[10] for surface operations scheduling by runway sequencing and push-back time control. These scheduling algorithms and tools were developed in a segregated fashion. For instance, TMA

*Research scientist, University Affiliated Research Center. Mail Stop 210-8. AIAA senior member
†Aerospace engineer, Aerospace High Density Operations Branch. Mail Stop 210-10. AIAA senior member

treats departure slots on runways as constraints when scheduling arrivals, whereas SARDA takes arrival times as hard constraints when sequencing departures. In these circumstances, runways are the competing resources for both departures and arrivals. In addition to runways, waypoints and route segments may also be shared between arrival and departure flows in terminal areas. Inefficient operations often emerge because of the constraints of shared resources. Recently, integrated schedulers[11–14] were proposed to apply speed controls and route options to optimize the schedules for both arrivals and departures with competing resources. Studies based on problems in San Jose,[11,12] Los Angles[13] , and New York[14] showed promise for improving operation efficiency in the presence of competing resources. On the other hand, benefits from optimal schedules calculated under deterministic scenarios are usually sensitive to uncertainties/errors of estimated arrival/departure times. A study conducted for a virtual single runway sequence optimization in the presence of uncertainties used a two-stage Mixed-Integer Linear Programming (MILP) formulation and Sample Average Approximation (SAA) by adjusting airspeed and push-back time.[15] For scheduling integrated operations using both route and speed controls, a stochastic scheduler based on Non-dominated Sorting Genetic Algorithm (NSGA) and Monte Carlo simulations was proposed to identify optimal and robust schedules.[16,17] The optimization algorithm in the scheduler takes uncertainty into account by calculating costs stochastically over thousands of possible estimated arrival/departure times that follow gaussian distributions. Another stochastic scheduler[18] that combines job-shop scheduling method and SAA was also proposed for solving integrated scheduling problems.

The scope of the aforementioned integrated schedulers, however, was small and only limited interaction points between certain departure and arrival flows were studied. A more integrated scheduler that coordinates arrivals, departures, and surface operations is necessary to provide more efficiency and/or even consider user preference by removing barriers between different operations.[19] This work develops a centralized stochastic scheduler for operations in a terminal area including airborne and surface operations on the basis of previous works[16,17] using NSGA and Monte Carlo simulations. It advances the sequential and stochastic scheduler developed previously and extends its application to arrivals, departures, and surface operations in the entire terminal area. In addition to the inclusion of a subset of competing waypoints between departures and arrivals, this work includes more competing resources between different flows, such as runway allocations (between departure and arrivals), departure fixes (among departures), and runway crossings (between departure and/or arrivals).

The Los Angeles (LAX) terminal area was used as an example and experiments were run with a four-hour traffic scenario in LAX. The scheduler was run sequentially to identify the best robust schedule for updated planning horizons. During the four-hour traffic scenario, schedules were updated periodically for each planning horizon. Final schedule solutions included the routes, speed or delays, and runway assignments. In the experiments, the standard deviation values of the departure time uncertainty were time-varied whereas the uncertainty means (arrival and departure) and arrival standard deviations were constant. In this paper, Section II describes the problem and its model. Section III presents the detailed methods proposed for this study. Results are provided and the analysis is presented in Section IV. Conclusions are provided in Section V.

II. Problem and Model

The interactions between Fillmore arrivals and Northbound departures in Los Angeles terminal airspace (Figure 1) have been investigated in previous works[13,16,17] on scheduling with competing resources. This work extends the scope to the entire LAX airspace by including all arrivals, departures and surface operations. There are typically more than 1,200 flights taking off and landing in a day at LAX, the research goal of this work is to develop an integrated and robust scheduler to statistically operate airborne and surface traffic efficiently.

Figure 1 presents the abstract model of LAX terminal airspace and surface/runway layouts used in this work. This layout models the West-flow configuration in LAX, which is reported to be the most common configuration with about 86% of the usage at LAX.[20] Arrival routes are in blue and departure routes are in green. The main differences from actual routes are the short-cut routes added for Fillmore arrivals and northbound departures to improve efficiency. Other than this, the model presented in Fig. 1 should be close to the actual West-flow configuration in LAX. There are eight entry points for arrival flights represented by incoming arrows to the green points. There are also four exit points for departures represented by outgoing arrows. On the surface, four runways 24L, 24R, 25L, and 25R are open for both arrivals and departures.

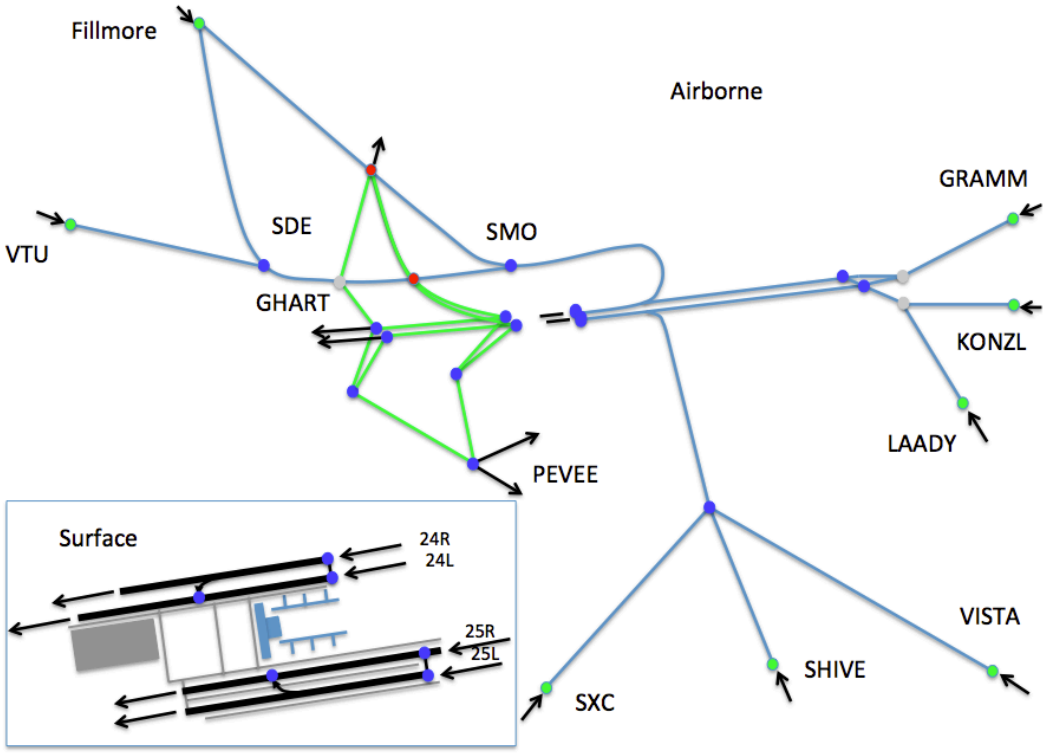

Figure 1. Los Angles Layout (not to scale)

The inner two runways 24L and 25R allow crossings from and to the outer runways through middle and end points. In this model, airborne flights are required to maintain minimum separations at blue, green, and red waypoints/fixes/nodes. Blue and green points are merging or diverging waypoints for traffic flows with the same direction, and red points are crossing waypoints between arrivals and departures. Grey points are either separated by altitudes or modeled as conflict-free points due to the immediate downstream constraints. On the ground, separation is required at departure and arrival nodes and crossing points on runways.

III. Method

In this section, a method of developing a centralized scheduler for operations in a terminal area including airborne and surface operations is discussed.

A. Objectives

The schedule optimization has multiple objectives with respect to total delay and pseudo controller intervention count. For each flight, the delay is defined as the extra flight transit time between entrance and exit points compared with shortest unimpeded transit time. The shortest unimpeded times assume flights take shortest routes at highest feasible speeds as if there are no other aircraft. Pseudo controller intervention count is the number of actions that need to be taken to avoid loss of separations due to prediction errors of flight arrival or departure times. When a flight needs to be delayed due to a separation violation at any merging and/or diverging waypoint including runways, a "controller action" is required to resolve the loss of separation, thus the pseudo controller intervention count will be increased by one.

$$\begin{cases} J_1 = \mathbb{E}[\sum_i t_i] - T_0 \\ J_2 = \mathbb{E}[\sum_i N_i] \end{cases} \tag{1}$$

The overall objectives are then to minimize expected values of total delays and controller intervention

counts, respectively. As shown in Eqn. 1, J_1 is the expected value of total delay over the random sampling space caused by uncertainties or prediction errors. The total delay can be expressed as the difference between the sum of flight transit times t_i for all flights and the sum of these flights' shortest unimpeded transit times T_0. As the sum of unimpeded transit times is a constant, its expected value is represented by T_0 in the equation. J_2 is the expected value of the sum of controller intervention counts for all flights. Both costs are evaluated over thousands sampling points using Monte Carlo simulations. Statistical measurements other than expected values can also be applied in the future if necessary.

B. Decision variables

There are eight arrival flows with eight respective entry points as shown in the Fig. 1. There are three departure flows corresponding to three directions: North, West, and South/East. Four types of decision variables are defined for every flight: delay, speed, route and runway. Delay may be applied before entry points. For a departure, the entry point is the gate on the surface, and for an arrival, it refers to the arrival meter fix in the airspace. For each flight segment, the speed is discrete and its feasible range is predefined based on preprocessed feasible speed calculated by the Trajectory Synthesizer (TS) tool used in the Center TRACON Automation System (CTAS).[21] Route options are designed for arrivals from Fillmore, and Northbound departures where the departures and arrivals can take short cut if conflicts don't happen at shared waypoints between these two flows. Four runways 24L, 24R, 25L, and 25R are available options to all flights, although factors like taxiing distance between runway and gate would affect such assignment.

C. Freeze horizon

The freeze horizon for arrivals is set at arrival meter fixes, which means that decision variables are frozen after arrival flights proceed beyond their entry points. For departures, the freeze horizon is set at the departing gates on the surface. A departure flight is kept at the gate until its decision variables including runway assignment, takeoff slot, and departing route are all decided. Once a departure flight leaves its gate, these assignments are then frozen. The freeze horizons are set up to avoid over-delay due to multiple rescheduling events. For instance, when a departure flight A leaves its gate, its schedule is frozen. If later on, another flight B is ready for scheduling and conflicts with A, then flight B has to yield to flight A in the updated schedule. Otherwise, if flight B overtakes flight A, then flight A can get further delayed. This situation may happen again if other flights cut in. Prior experiments showed that flight A can eventually get significant delay if left unfrozen.

D. Constraints

As the aircraft speed options are preprocessed, infeasible speed options are ruled out. Separation requirements are the only constraints in the work. They were applied at fixes, waypoints, and runways, where flights merged and/or diverged. According to FAA's regulations based on wake vortex,[22] minimum separation requirements for airborne flights were listed in Table 1. The separation for departure and crossing flights at the runway is given in Table 2 by converting original distance separation into time-based separation.[23]

Table 1. Minimum Airborne Aircraft Separations

Separation Distance (nmi)		Leading Aircraft: Wake Category			
		Heavy	B757	Medium	Small
Trailing Aircraft Wake Category	Heavy	4	4	3	3
	B757	4	4	3	3
	Medium	5	4	3	3
	Small	6	5	3	3

In this optimization, the separation constraints are not treated as "hard constraints" or inequalities as in usual linear or nonlinear optimization formulations. Because the same amount of separation violation results in different delays in air traffic system, in this model, delay costs were calculated for resolving separation

Table 2. Minimum Runway Separations for Departure and Crossing Flights

Separation Time (sec)		Leading Aircraft Wake Category				
		Small	Large	Heavy	B757	Crossing
Trailing Aircraft Wake Category	Small	59	88	109	110	25
	Large	59	61	109	91	25
	Heavy	59	61	90	91	25
	B757	59	61	109	91	25
	Crossing	40	40	40	40	40

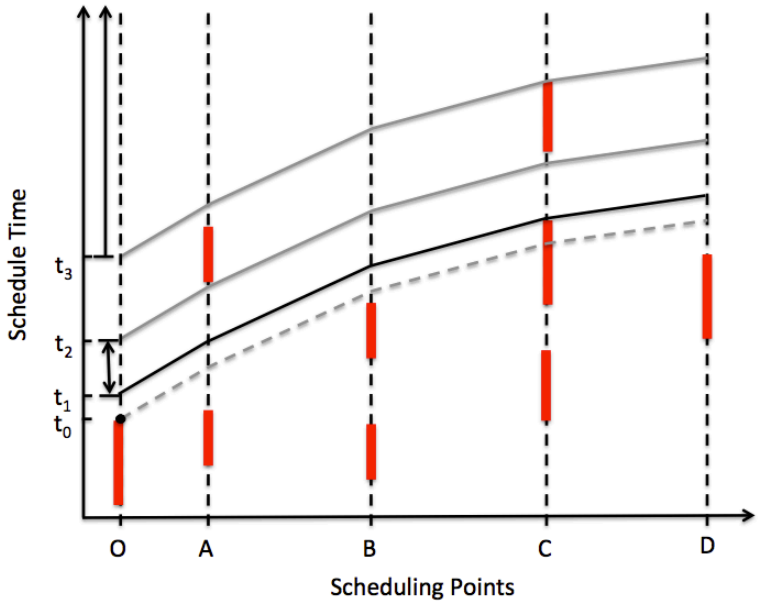

Figure 2. Conflict-free scheduling using Constraint Algebra

violations. Given decision variables, a method based on constraint algebra[24] was used to resolve aircraft conflicts or constraint violations. With final resolutions from this method, arrival times at exit points were then used to calculated delay costs. In the constraint algebra method, the flight priority is set up according to runway unimpeded estimated times of arrival (ETAs) at the runway for both departures and arrivals. The basic idea of this method is to insert one flight at a time starting from the first flight on the priority list. The insertion should be guaranteed conflict-free at all points. The speed change is not considered in the constraint algebra method, because it is already included in decision variables. If there is any conflict between the inserting flight and existing flights, the new flight is given extra delay at its starting point. For a departure flight, the delay is imposed at its gate as extra gate waiting time. Whereas, for an arrival flight, the delay is propagated back upstream, beyond the arrival meter fix. Figure 2 presents an example of inserting a flight using constraint algebra. Red bars represent the slots occupied by other aircraft. Given a speed profile, the relative time differences between scheduling points are fixed for this flight. The constraint algebra method calculates open slots at each scheduling point, and then finds the feasible starting time slots with the "rigid" speed profile. In this figure, the feasible ranges for inserting the given speed profile are from t_1 to t_2 and any time after t_3. Therefore, the extra delay cost of inserting this flight is $t_1 - t_0$, which would be included in the final delay cost. As the speed is treated as a decision variable, the scheduler will calculate the delay costs for a range of speed profile options in this way during optimization.

E. Uncertainty models

In TMA and SARDA, schedules are updated frequently (every 10 seconds) to mitigate the impact of uncertainties/prediction errors. When routes and/or runway assignments need to be decided in schedules for both arrivals and departures, they have to be decided with a certain look-ahead time of at least several minutes. For example, when a departure flight is about to taxi from its gate, pilots need to know the designated runway. To determine the runway, uncertainties in other flights' arrival or departure times have to be taken into account because it would not be feasible to change its runway assignment in next update cycle even if prediction errors are found later. In this circumstance, increasing update frequency is not effective for mitigating the impact of uncertainties. Taking uncertainties directly into account at planning stages is helpful. Arrivals uncertainties must also be taken into account prior to a freeze horizon as runway/route is decided ahead of time.

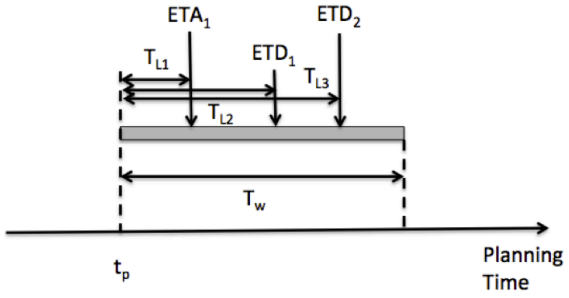

Figure 3. Definition of the look-ahead time

$$\begin{cases} \sigma = 0.41 \times T_L, & T_L \text{ is look-ahead time} \\ \mu = C, & \text{C} = 30.0 \end{cases} \tag{2}$$

To simulate the real world, this study assumes that "planned times" to entry points are uncertain and follow certain distributions. For a departure flight, the "planned time" is the planned push-back time from its gate. And for an arrival flight, it refers to the estimated arrival time to the arrival meter fix. The ETA of an arrival to entry points is assumed to follow a normal distribution with zero means and 30 second standard deviation regardless of the planning look-ahead times based on trajectory prediction studies.[25] Whereas for a departure, it is defined that the estimated time of departure (ETD) from a gate, or the push-back time, follows a normal distribution with a standard deviation linearly increased with the look-ahead time T_L where the mean is still constant (see Eqn. 2). This definition and coefficients are set up based on study results from the Surface Decision Support System (SDSS).[26,27] The definition of the look-ahead time T_L is described in Fig. 3, in which the ETA_1, ETA_2, and ETD_1 are ETAs and ETDs for sample flights with in a time window span of T_w. The T_L of each flight is the time differences between its ETA/ETD and the planning time t_p. According to the equation, when the planning look-ahead time increases, the uncertainty of departure time increases. Therefore, ETD_2 is more uncertain than ETD_1.

F. Scheduling Scheme

To generate a schedule for a traffic scenario with a large time window, the most practical and efficient approach is to divide it into smaller windows. The relation between planning horizon and actual time is shown in Fig. 4. A sequential dynamic scheduling scheme is adopted, where the schedule for each window with time span T_w is identified before the time proceeds to that window at t_1. If the computational time of the scheduling algorithm T_c is considered, the planning should be started at or before $t_1 - T_c$ to make sure an updated schedule is available at actual time t_1. This process will be repeated periodically to provide schedules for all time windows. An overlapping scheduling scheme is similar except that planning horizons overlap by a certain duration.

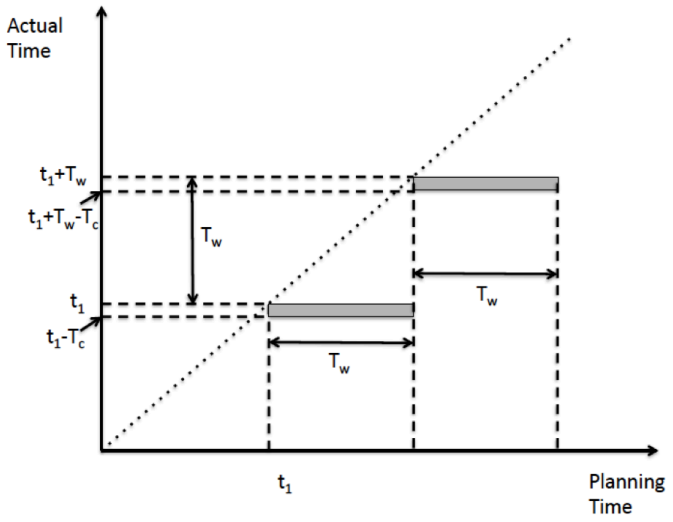

Figure 4. Scheme with nonoverlapping scheduling windows

G. Algorithms and implementation

The NSGA, a variant of Genetic Algorithms, is applied to solve this multi-objective optimization problem due to its promising capability of handling multiple objectives and nonlinear optimizations. The stochastic scheduler combines NSGA and Monte Carlo simulation.[13,16,17,28] The decision variables including speeds, routes, delays, and runway assignment are coded as "genes", and each solution with a set of decision variables is marked as an "individual". In NSGA, a population with hundreds of "individuals" evolves at each generation in terms of their costs through operations of "crossover", "mutation", "ranking", and "selection". When evaluating costs for each solution or "individual", Monte Carlo simulations are used to identify the statistical measurements. Given a solution with a set of decision variables, its costs are evaluated thousands of times using the constraint algebra in Monte Carlo simulations. In each simulation, the computed costs correspond to one uncertainty sample, which was imposed on the "planned" push-back or arrival times. On top of the optimization core code, scripts that involve the freeze horizon and planning horizon set-ups are then applied to identify outputs for current planning horizon and inputs for next one. This approach is implemented using CUDA programming with GPUs to reduce the significant computational time to a reasonable level. In a Linux platform with 18x2.5 GHz Xeon, 32 GB memory, and two GeForce GTX690 GPUs, a 15-flight scenario scheduling problem takes around 30 seconds to be solved .[28,29]

IV. Results

A four-hour traffic scenario was built based on historical traffic on Jul. 1, 2014. A total of 315 flights were included composed of 172 departures and 143 arrivals for LAX. In this scenario, information of each flight includes aircraft type, planned time at entry points, gate assignment, and flight routes including arrival/departure meter fixes. Various experiments were then conducted to examine the impacts of difference factors and to analyze the potential benefits introduced by those factors.

A. Planning horizon

In deterministic cases, or stochastic cases with constant means and constant standard deviations, the delay of the best solution decreases when the window size/planning horizon increases. Longer planning horizon helps find solutions closer to global optimum because more information is involved. This may not always be true when means and standard deviations become time-dependent, or in other words, when uncertainties increase with the look-ahead time. Since uncertainties grow in long planning horizons due to long look-ahead times, the increased uncertainties diminish the benefits introduced by expended knowledge because the predicted information becomes useless or even harmful when the look-ahead time is long.

Figure 5 shows the minimum delays that can be achieved by solutions generated under different planning

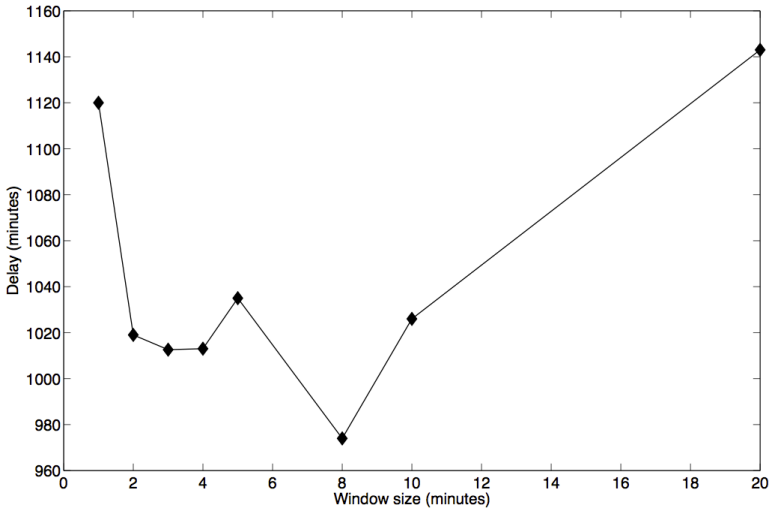

Figure 5. Minimum delays corresponding to various planning windows

horizons. It shows that solutions with smaller planning windows produce less delay when the planning horizon decreases from 20 minutes to eight minutes. This demonstrates that the impact of time-varied uncertainties abate the advantage introduced by long planning horizons. The delay cost increases when the planning window is reduced further. Eventually it returns to a relatively high level at a one-minute planning window. Because the scheduler cannot foresee flights coming in next planning window, the scheduler can only make decisions based on knowledge in the current window. When the planning horizon is too short, the knowledge becomes too limited for the optimizer to generate a good solution.

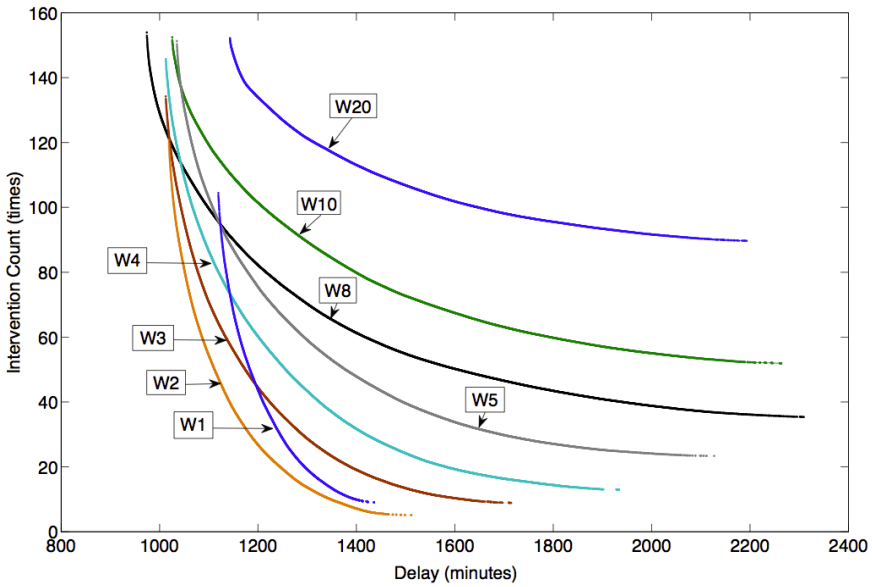

Figure 6. Pareto fronts for different planning windows

The optimizer tries to minimize two costs in the model. Figure 6 shows cost pareto fronts produced by the optimization associated with different planning windows. Each pareto front is composed of a set of points with each point referring to a schedule solution. The coordinates of a point represent two statistical costs associated with the solution as mentioned in previous sections: the first cost is the total delay including airborne and ground delays; And the second cost is the pseudo controller workload (intervention count). The notion of "W" in the figure denotes planning window. For instance, "W20" denotes solutions generated with a 20-minute planning window.

Trends observed in Fig. 6 are similar to Fig. 5. When the planning horizon decreases from 20 minutes to eight minutes, small windows outperform large windows along the entire pareto front. Trade-offs rise when the window size decreases from eight minutes to two minutes. Figure 6 shows that with an eight-minute planning window, the scheduler produces solutions with the minimum delay. The duration of planning horizon that produces minimum delay can be affected by several factors: freeze horizon set-up, taxiing time from gate to runway, and relation between uncertainties and time.

Here is an example used to compare solutions from eight-minute planning horizon and two-minute planning horizon: *A departure Flight A with type A321 is at Gate 42B and according to its plan, it will leave the gate at 08:11:00. Flight B is B737 located at Gate 7. Its planned push-back time is 08:13:30. The taxiing distances from Gate 42B to runway 24L, 24R, 25R, and 25L are 1.3 nmi, 1.42 nmi, 1.32 nmi, and 1.44 nmi, respectively. The corresponding distances from Gate 7 to runways are 0.1 nmi, 0.22 nmi, 2.6 nmi, and 2.72 nmi, respectively. Current scheduling time is at 08:10:00.*

For a two minute planning horizon, at 08:10:00, the scheduler did not know any information after 08:12:00 including Flight B. Therefore, the scheduler allowed Flight A to leave the runway when it was ready, and assigned runway 24L to the flight, which takes Flight A 312 seconds to taxi from the gate to runway. Then Flight A entered the frozen horizon right after pushing back from gate 42B. When the time proceeded to next planning horizon at 08:12:00, the scheduler finds Flight B, which would be ready in 90 seconds. Its taxiing time to 24L is 24 seconds. Because Flight A was in frozen horizon already, the only option available to Flight B was to wait and follow Flight A, which cost Flight B 169 seconds in waiting time. Whereas, for an eight minute planning horizon, Flight B was within scope when the scheduler was calculating at 08:10:00. Although there was 210 seconds look-ahead time with 86 second standard deviation for Flight B, the scheduler decided to put Flight B in front of Flight A based on statistical cost measurements, which would reduce delay about 160 seconds.

Apparently, under current uncertainty assumption, the eight-minute planning horizon is the best candidate if a solution with minimum delay is desired. When the requirement is relaxed by allowing a bit high delay, the scheduler with a two-minute planning horizon can also be used as it can generate solutions with a bit high delay but much lower intervention count. This implies that, in application, any planning horizon from two minutes to eight minutes could be a good candidate. One extreme example is the one-minute planning window, the delay level of the pareto front becomes much greater than other short planning horizons. The hypothesis is that, although time-dependent uncertainties favor short planning horizon, one minute is too short for the given freeze horizon and taxiing time. The lack of flight information in the next planning horizon leads the optimizer to generate solutions with high delay levels.

B. Stochastic vs. deterministic

To examine the difference between deterministic and stochastic schedulers, experiments using a deterministic scheduler were conducted. To mimic the time-dependent uncertainties in deterministic experiments, time-dependent errors/noises were imposed to the "planned time" of each flight that is involved in the planning horizon. The deterministic optimization is then conducted on the basis of these noisy "planned times" to generate schedules. A variety of planning strategies were tested by varying planning horizons and update cycles. For example, the black square in Fig. 7 denotes the solution produced by the deterministic scheduler with a 15-minute planning horizon and one minute update cycle (marked as "W15U1"). This means the optimizer calculates or updates schedule every minute for the following 15-minute planning horizon. For the sake of comparison, the pareto front generated under the eight-minute planning window using the stochastic scheduler (shown in Fig. 6) is presented as a black curve in Fig. 7.

This figure shows clearly that the stochastic scheduler outperforms the deterministic scheduler based on the statistical measurements of both delays and pseudo intervention counts. This conclusion holds for deterministic schedulers with all strategies: overlapped or non-overlapped planning windows, and long or short planning horizons. With the same level of intervention counts, the stochastic scheduler reduces delays anywhere between 28% and 40% when comparing against the deterministic scheduler. The imperfect knowledge of "planned time" and neglect of uncertainty in optimization appear to be the two main factors that contribute to the differences. In deterministic optimization, every piece of information is assumed to be perfect and the evolution at every step in the optimization is built on that assumption, apparently, inaccurate "planned times" can not lead the optimizer to the optimal solution. Whereas, in stochastic optimization,

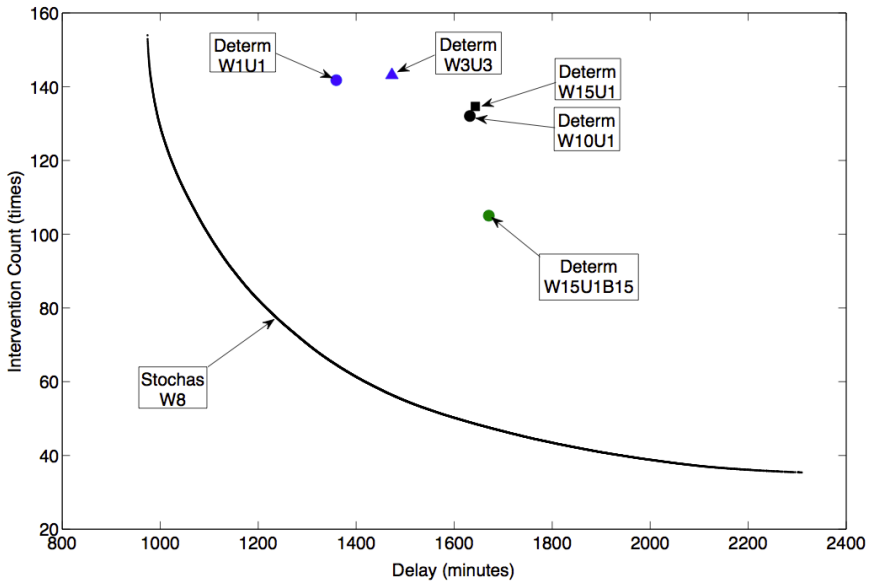

Figure 7. Comparison between stochastic and deterministic schedulers

the scheduler assumes that "planned time" is imperfect. By following a certain distribution, the scheduler finds the optimal solution based on statistical measurements.

A typical way to deal with uncertainty in deterministic optimization is to impose extra buffers besides basic separation requirements, the green circle in Fig. 7 shows the final solution from a deterministic optimization using a 15-minute planning horizon and one-minute update cycle with a 15-second buffer. The 15-second buffer does reduce the intervention count a lot, however, it also increases the total delay. This implies that stochastic scheduling is an effective way to reduce delay and intervention count statistically, since it takes uncertainty knowledge into account during the optimization.

C. Uncertainty Magnitude

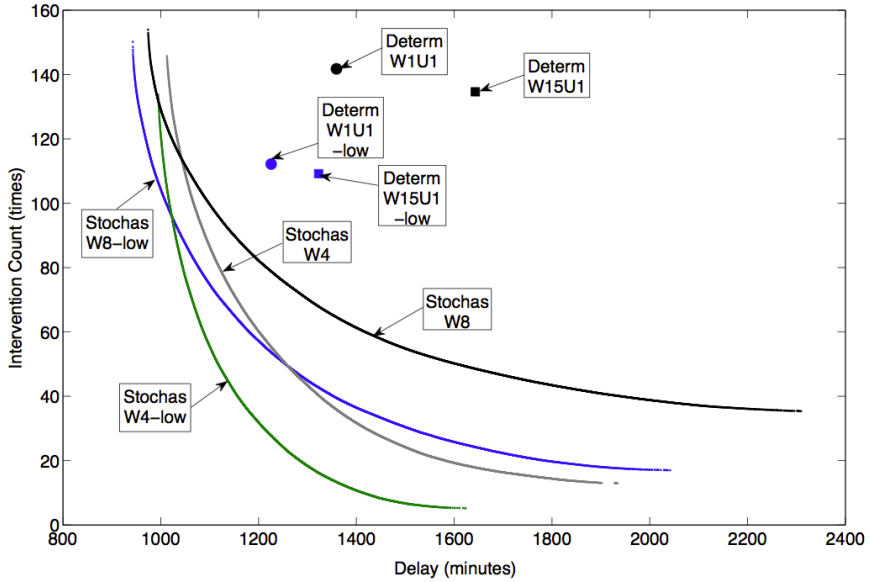

Figure 8. Impact of Uncertainty Magnitude

To examine the impact of uncertainty magnitude on final schedules, the linear coefficient 0.41 in Eqn. 2

is reduced to 0.2. Figure 8 presents the results with this lower uncertainty. The labels containing "low" refer to results with low uncertainties. As expected, with more accurate "planned times" both stochastic and deterministic schedulers generated solutions with lower delays. Results from deterministic schedulers improved more than stochastic schedulers, which indicated that deterministic schedulers were more sensitive to uncertainties. In stochastic cases, the big planning horizon benefit more from the accuracy improvement than the small planning horizon. This experiment denotes that the advantage of stochastic schedulers over deterministic schedulers is closely correlated to the uncertainty magnitude.

D. Runway usage

Runway makespan and occupancy metrics were used to examine runway usage. Runway makespan refers to the time span between first and last flight for a set of operations. When flights are not operated back-to-back all of the time, the correlation between the makespans and the effectiveness of schedules may not be tight. However, it is nevertheless a good metric to show the effectiveness of schedules. Figure 9 presents the statistical runway makespans corresponding to three difference schedulers: a stochastic scheduler with an eight-minute planning horizon, a deterministic scheduler with a 15-minute planning horizon with one-minute update cycle, and a deterministic scheduler with a three-minute planning horizon. It shows that the stochastic scheduler produced minimum makespans for all four runways. Compared to the deterministic scheduler with a 15-minute window, the stochastic scheduler finished all flight operations six minutes earlier on runway 24L and 10 minutes earlier on runway 25R.

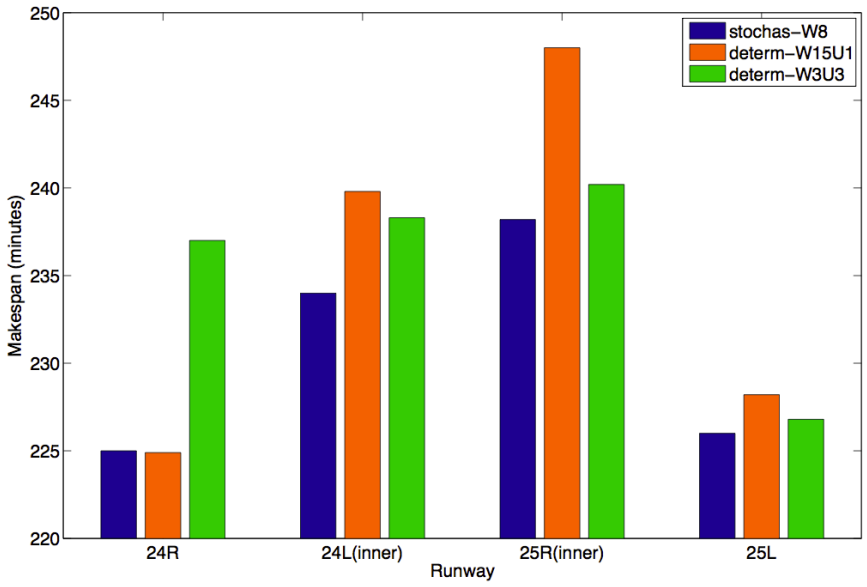

Figure 9. Runway Makespans

Runway occupancy can serve as another metric to examine the characteristics of runway usage. It is defined as the total percentage of the usage of a runway. When computing the metric, it is assumed that each departure flight uses 40 seconds to takeoff from a runway and each arrival flight takes 50 seconds to land on a runway. Crossing a runway is assumed to take 10 seconds for a flight. Figure 10 presents the statistical runway occupancies corresponding to the same three difference schedulers presented in Fig. 9. Figure 10 shows that all three schedulers utilized 25L/R runways more than 24L/R runways. This could be caused by the nature of the scenario. Flights were typically assigned to the nearest runway in most situations and there are more flights close to 25L/R runways in the experimental scenario. Because all flights have to utilize runway to arrive or to depart, the overall number of operations should be similar with some difference existing in the number of crossings only. It is noted that outer runways were used a bit more in stochastic cases, which implies that there were more runway crossings in stochastic cases than in deterministic cases. The fact that the occupied time of both inner runways in the stochastic case is less than deterministic cases indicates that more departures were assigned to inner runways in the stochastic case. The overall occupancies of the three schedulers are similar, however, the stochastic scheduler occupancies are slightly lower, which

indicates that schedules from stochastic optimization utilize runways slightly more efficiently.

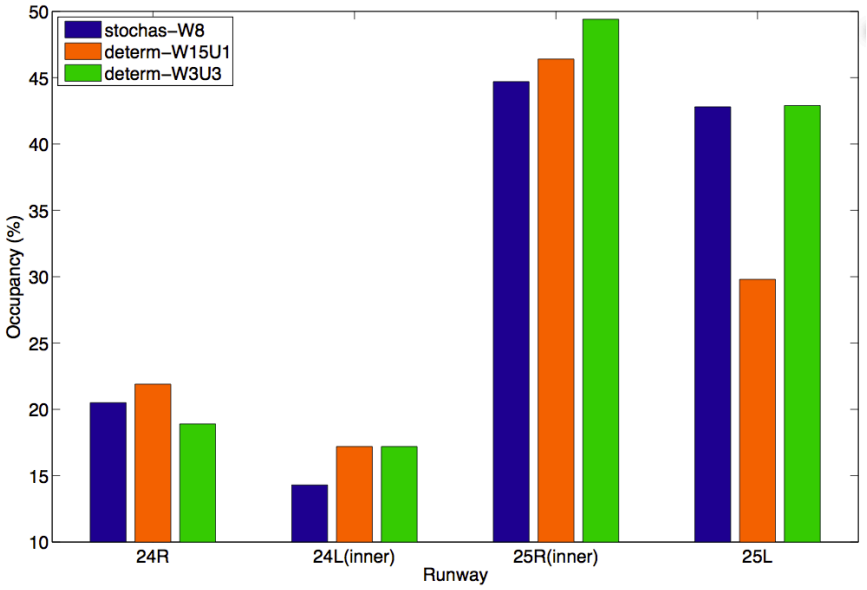

Figure 10. Runway Occupancy

V. Conclusions

An integrated scheduler that coordinates arrivals, departures, and surface operations provides efficiency in terminal areas by removing barriers between different operations. This work developed a centralized stochastic scheduler for operations in a terminal area including airborne and surface operations based on NSGA and Monte Carlo simulations methods. It extended the formulation of the sequential, stochastic, and integrated scheduler to arrivals, departures, and surface operations in an entire terminal environment. In addition to the competing waypoints, this extension included most competing resources in terminal areas between different flows, such as runway allocations, departure fixes, and runway crossings.

The Los Angeles terminal area was used as an example and experiments were run with a four-hour traffic scenario in LAX. The scheduler was run sequentially to identify the best robust schedule for the next planning horizon. In the experiments, the standard deviation values of the departure time uncertainty were time-varied whereas the departure and arrival uncertainty means and arrival standard deviations were constant. Final schedules included decisions on routes, speeds or delays, and runway assignments. The algebra constraint method was used to calculate extra costs to satisfy separation requirements, which were eventually amended to the overall costs. Studies on planning horizons showed that trade-offs exist between planning horizons and achievable minimum delays. A 20-minute planning horizon was not a good choice because uncertainties grew with the look-ahead time. Eight minutes was promising for planning as it achieved the lowest delay compared to others. However, the results demonstrated that any duration from two minutes to eight minutes could be a good candidate as well. Experimental results showed that using stochastic schedulers reduced the flight time delay (airborne and ground) 28% to 40% statistically compared to deterministic schedulers with the same level of intervention counts. Experiments on uncertainty magnitude demonstrated the close correlation between uncertainty and the benefit from stochastic schedulers. It was also shown that deterministic schedulers are more sensitive to uncertainty than stochastic schedulers. The results on runway usage showed that using the stochastic scheduler, runway makespans and occupancy were usually slightly lower when compared with deterministic schedulers. Overall, experiments showed that this sequential stochastic scheduler was capable of scheduling arrivals, departures, and surface operations in an integrated fashion. The stochastic scheduler successfully took uncertainty into account and statistical results showed significant delay savings were achieved when the knowledge of uncertainties was involved.

References

[1]Dear, R. G., "The Dynamic Scheduling of Aircraft in the Near Terminal Area," Tech. Rep. Flight Transportation Laboratory Report R76-9, MIT, September 1976.

[2]Neuman, F. and Erzberger, H., "Analysis of Delay Reducing and Fuel Saving Sequencing and Spacing Algorithms for Arrival Spacing," Tech. Rep. NASA Technical Memorandum 103880, October 1991.

[3]Dear, R. G. and Sherif, Y. S., "An Algorithm For Computer Assisted Sequencing and Scheduling of Terminal Operations," *Transportation Research A*, Vol. 25A, No. 2/3, 1991.

[4]Gupta, G., Malik, W., and Jung, Y. C., "A Mixed Integer Linear Program for Airport Departure Scheduling," *9th AIAA Aviation Technology, Integration and Operations Conference (ATIO)*, Hilton Head, South Carolina, 21-23 September 2009.

[5]Montoya, J., Wood, Z., Rathinam, S., and Malik, W., "A Mixed Integer Linear Program for Solving a Multiple Route Taxi Scheduling Problem," *AIAA Guidance, Navigation, and Control Conference*, Toronto, Canada, 2-5 August 2010.

[6]Rathinam, S., Wood, Z., Sridhar, B., and C., J. Y., "A Generalized Dynamic Programming Approach for a Departure Scheduling Problem," *AIAA Guidance, Navigation, and Control Conference*, Chicago, IL, 10-13 August 2009.

[7]Balakrishnan, H. and Chandran, B., "Scheduling Aircraft Landings Under Constrained Position Shifting," *AIAA Guidance, Navigation, and Control Conference*, Keystone, CO, 21-24 August 2006.

[8]Swenson, H., Hoang, T., Englland, S., Vincent, D., Sanders, T., Sanford, B., and Heere, K., "Design and Operational Evaluation of the Traffic Management Advisor at the Fort Worth Air Route Traffic Control Center," *1st USA/Europe Air Traffic Management Research and Development Seminar*, Saclay, France, 17-19 June 1997.

[9]Swenson, H., Thipphavong, J., Sadovsky, A., and Chen, L., "Design and Evaluation of the Terminal Area Precision Scheduling and Spacing System," *Ninth USA/Europe Air Traffic Management Research and Development Seminar*, Berlin, Germany, 2011.

[10]Jung, Y., Huang, T., Montoya, J., Gupta, G., Malik, W., and Tobias, L., "A Concept and Implementation of Optimized Operations of Airport Surface Traffic," *10th AIAA Aviation Technology, Integration and Operations Conference (ATIO)*, Fort Worth, TX, September 13-15 2010.

[11]Capozzi, B. J. and Atkins, S. C., "Towards Optimal Routing and Scheduling of Metroplex Operations," *9th AIAA Aviation Technology, Integration, and Operations Conference(ATIO)*, Hilton Head, South Carolina, 21-23 September 2009.

[12]Capozzi, B. J. and Atkins, S. C., "A Hybrid Optimization Approach to Air Traffic Management for Metroplex Operations," *10th AIAA Aviation Technology, Integration and Operations Conference (ATIO)*, Fort Worth, Texas, 13-15 September 2010.

[13]Xue, M. and Zelinski, S., "Optimal Integration of Departure and Arrivals in Terminal Airspace," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 1, 2014.

[14]Chen, H. and Zhao, Y. J., "Sequential Dynamic Strategies for Real-Time Scheduling of Terminal Traffic," *AIAA Journal of Aircraft*, Vol. 49, No. 1, 2012.

[15]Solveling, G., Solak, S., Clarke, J. P., and Johnson, E., "Runway Operations Optimization in the Presence of Uncertainties," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 5, 2010.

[16]Xue, M. and Zelinski, S., "Optimization of Integrated Departures and Arrivals Under Uncertainty," *13th AIAA Aviation Technology, Integration, and Operations Conference(ATIO)*, Los Angeles, LA, 12-14 August 2013.

[17]Xue, M. and Zelinski, S., "Dynamic Stochastic Scheduler For Integrated Arrivals and Departures," *33rd Digital Avionics Systems Conference (DASC)*, Colorado Springs, CO, 5-9 October 2014.

[18]Bosson, C., Xue, M., and Zelinski, S., "Optimizing Integrated Terminal Airspace Operations Under Uncertainty," *33rd Digital Avionics Systems Conference (DASC)*, Colorado Springs, CO, 5-9 October 2014.

[19]Zelinski, S., "A Framework for Integrating Arrival, Departure, and Surface Operations Scheduling," *33rd Digital Avionics Systems Conference (DASC)*, Colorado Springs, CO, 5-9 October 2014.

[20]Timar, S. D., Nagle, G., Saraf, A., Yu, P., Hunter, P., Trapani, A., and Johnson, N., "Super Density Operations Airspace Modeling for the Southern California Metroplex," *AIAA Modeling and Simulation Technology Conference*, Portland, Oregon, 08-11 August 2011.

[21]Erzberger, H., Davis, T. J., and Green, S. M., "Design of Center-TRACON Automation System," *ARARD Meeting on Machine Intelligence in Air Traffic Management*, Berlin, Germany, May 11-14 1993.

[22]*Aeronautical Information Manual (AIM): Official guide to basic flight information and ATC procedure*, Federal Aviation Administration (FAA), 2014.

[23]Gupta, G., Malik, W., and Jung, Y. C., "Incorporating Active Runway Crossings in Airport Departure Scheduling," *AIAA Guidance, Navigation, and Control Conference*, Toronto, Canada, 2-5 August 2010.

[24]Meyn, L., "A Closed-Form Solution to Multi-Point Scheduling Problems," *AIAA Modeling and Simulation Technologies Conference*, Toronto, Ontario, Canada, 2-5 August 2010.

[25]Stell, L., "Prediction of Top of Descent Location for Idle-thrust Descents," *The Ninth USA/Europe Air Traffic Management Research and Development Seminar*, Berlin, Germany, June 2011.

[26]Capps, A. and Engelland, S., "Characterization of Tactical Departure Scheduling in the National Airspace System," *11th AIAA Aviation Technology, Integration and Operations Conference (ATIO)*, Virginia Beach, VA, September 20-22 2011.

[27]Atkins, S., Jung, Y., Brinton, C., Stell, L., and Rogowski, S., "Surface Management System Field Trial Results," *4th AIAA Aviation Technology, Integration and Operations Conference (ATIO)*, Chicago, IL, September 20-22 2004.

[28]Xue, M. and Zelinski, S., "Integrated Arrival and Departure Schedule Optimization Under Uncertainty," *AIAA Journal of Aircraft*, 2015 (accepted).

[29]Bosson, C., Xue, M., and Zelinski, S., "GPU-based Parallelization for Schedule Optimization with Uncertainty," *AIAA Aviation and Aeronautics Forum and Exposition*, Atlanta, GA, June 16-20 2014.